32
150 pts.

RUN

Enciclopedia Práctica del Spectrum

Nueva Lente/Ingelek

GUARDANDO DATOS

Hasta ahora, hemos tratado sobre el conjunto de sentencias que nos permiten efectuar los mismos tipos de almacenamientos de programas y datos ya familiares en el empleo del casete. Es decir, tanto el casete como la unidad MICRODRIVE nos permiten almacenar programas BASIC, bloques de memoria, pantallas y matrices de datos.

Sin embargo, no hemos hablado de una nueva posibilidad que nos brinda en exclusiva el empleo de la unidad MICRODRIVE: la creación y mantenimiento de ficheros de datos.

Por fichero, dentro de la jerga informática, se entiende un conjunto de registros relacionados entre sí, guardando un determinado orden y conteniendo información del mismo tipo. Esta información se encuentra dividida en campos, a su vez compuestos por una serie de caracteres (bytes).

A pesar de ser el CARACTER la mínima unidad en que podemos dividir un registro, es el CAMPO la verdadera unidad elemental de información, al ser la porción más pequeña de REGISTRO que alberga para nosotros un significado concreto.

Dentro de esta estructura tenemos amplia libertad de elección, ya que un fichero puede albergar un número variable de registros, cada registro un número variable de campos y, por último, cada campo un número variable de caracteres.

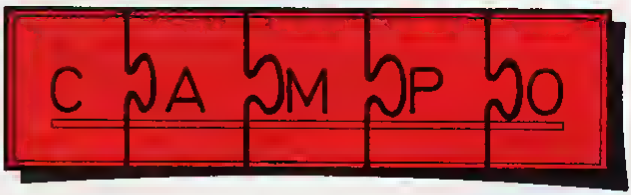
Sin duda veremos esto más claro con un ejemplo. Imaginemos que deseamos construir un archivo (fichero) capaz de funcionar como una agenda telefónica. Para ello, debemos estudiar primero los datos (campos) con los cuales vamos a trabajar. Nos hará falta el nombre de cada persona, su dirección y número de teléfono.

Atendiendo a estas necesidades debemos pensar en un fichero de agenda telefónica el cual contendrá determinado número de registros, con tres campos en cada uno de ellos, de longitud variable.

En el lenguaje informático se le da el nombre de secuencial al tipo de fichero que soporta la unidad MICRODRIVE. Aclararemos que se entiende por fichero secuencial, aquel en el cual es preciso efectuar la lectura de todos los registros anteriores para tener acceso a uno determinado.

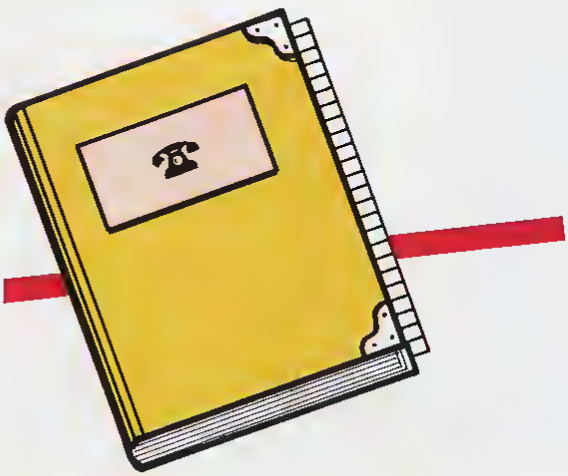
Volviendo a nuestro ejemplo anterior, supongamos que hemos grabado un fichero con el nombre, dirección y teléfono de nuestros amigos, de forma que los cuatro primeros registros corresponden a los datos de Luis, Juan, Roberto y Enrique, respectivamente. Pues bien, si queremos recuperar de esta información así grabada algún dato de Roberto, deberemos leer obligatoriamente antes los de Luis y Juan, aunque no nos interesen, antes de poder acceder a los correspondientes a Roberto.

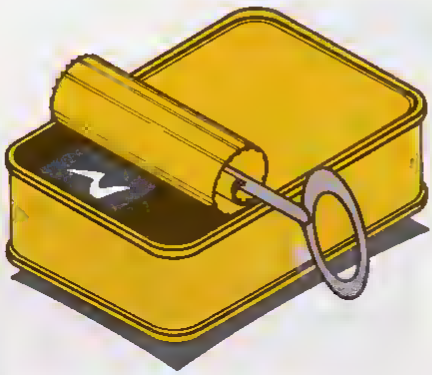
Dicho de otra forma, los registros almacenados en el fichero están en el mismo orden en que se



Un campo es la unidad elemental de información que alberga un significado concreto

Podemos citar como ejemplo práctico de archivo un programa que sirva de agenda telefónica





Los ficheros de datos han de abrirse antes de hacer uso de ellos, tanto para la lectura como escritura.

¡!

El *buffer* es una zona de almacenamiento temporal de datos donde éstos se van colocando hasta agotar la capacidad total disponible de 512 bytes. Una vez completada esta los datos se graban en bloque en el fichero abierto

*

Un fichero de organización secuencial es aquel en el cual es preciso efectuar la lectura de todos los registros anteriores para tener acceso a uno determinado

produce la entrada de datos. Por ello, debemos poner especial cuidado en que la operación de grabación del fichero se efectúe en el orden que realmente nos interesa, puesto que esta decisión influirá más tarde en la eficacia a la hora de recuperar los datos.

APERTURA Y CIERRE DE FICHEROS

Los ficheros de datos han de abrirse antes de hacer uso de ellos tanto para lectura como escritura, así como cerrarse una vez concluida la consulta o actualización de los mismos. En este sentido, existe una gran similitud entre lo que sucede con un fichero de clientes en una oficina y uno en MICRODRIVE.

Supongamos que en un archivador metálico (fichero) se encuentran las carpetas (registros) de cada uno de los clientes, en cuyo interior se han escrito sus nombres y direcciones (campos). Si deseamos acceder a este archivo debemos primero abrirlo con la llave, para localizar la ficha (registro) que nos interesa, basándonos para ello en el orden en que se encuentran las fichas, y efectuar la consulta o modificación de ésta. Concluida la operación, colocaremos la ficha nuevamente en su sitio para cerrar el archivo con llave.

Del mismo modo, cuando tratamos con ficheros en MICRODRIVE es necesario pasar por tres fases: apertura, actualización y cierre del archivo. La fase de cierre, como explicaremos más adelante, es tan importante como la de apertura puesto que, si bien es cierto que sin abrir el fichero no tenemos acceso a el, no lo es menos que si olvidamos cerrarlo al concluir la actualización, pues perderemos los datos grabados.

APERTURA DE FICHEROS

La apertura y asignación de nombre y número de canal a un fichero en MICRODRIVE se produce, conjuntamente, a través de la sentencia **OPEN #**. El formato general de la sentencia es

OPEN #c; "M";n,"nombre-fichero"

Donde **c** hace referencia al número de canal por el cual circularán los datos (0-15), **M** al tipo de

Se llama secuencial, al tipo de fichero que soporta la unidad MICRODRIVE

OPEN

La apertura y asignación de nombre y número de canal en MICRODRIVE, se produce a través de OPEN #.

unidad que almacenará el fichero (MICRODRIVE), **n** al número del MICRODRIVE (1-8) y **nombre fichero** a la cadena alfanumérica de hasta 10 caracteres de longitud que compone el nombre del fichero dentro del catálogo del MICRODRIVE

Llegados a este punto, debemos aclarar que el Spectrum dispone de 16 canales (0-15) por los cuales puede circular la información de forma independiente. Esto permite que se produzcan entradas y salidas desde y hasta diferentes periféricos al mismo tiempo. Por ejemplo, podemos estar leyendo datos de dos ficheros en MICRODRIVE, grabándolos sobre otros dos y efectuando una salida a impresora y pantalla de los contenidos de los registros simultáneamente.

Aunque en principio podríamos asignar a un fichero cualquier número de canal, no es recomendable el uso de los números 0 y 1 para la salida de datos hacia la línea 23 de la pantalla, así como para la entrada de caracteres desde el teclado. Del mismo modo, los canales 2 y 3 son empleados para producir las salidas de la sentencia **PRINT** a la parte superior de la pantalla, y para la sentencia **LPRINT** en impresora respectivamente.

Además, la asignación de número de canal a un fichero de datos en el momento de su apertura permite, en lo sucesivo, referirse a él para las operaciones de lectura o escritura con sólo hacer uso de su número, obviando el nombre.

Lo primero que comprueba la sentencia **OPEN #** es la existencia en el cartucho de un archivo con ese nombre. Caso de existir, interpreta que se van a realizar operaciones de lectura del mismo (consulta). En caso contrario, interpreta que debe abrir un fichero con ese nombre para efectuar operaciones de escritura (actualización).

Ademas, si en su búsqueda encuentra un nombre igual en el cartucho correspondiente a un programa, bloque de memoria, pantalla o matriz de datos, emite el mensaje de error *Wrong file type* (tipo de fichero equivocado).

GRABACION DE DATOS

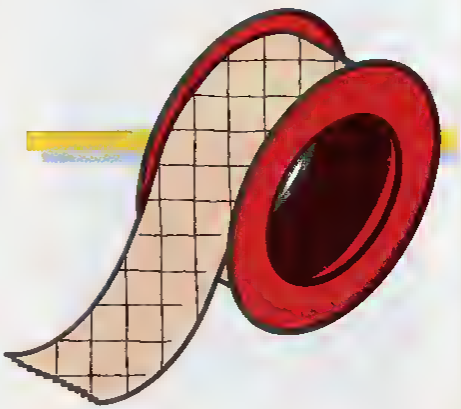
La grabación de datos en cartucho se efectúa por medio de la sentencia **PRINT #**. Esta sentencia permite la escritura en cartucho de forma similar a como lo hace en pantalla su homóloga **PRINT**, con la particularidad de ser necesario especificar a continuación el número de canal por el que escribir.

Las posibilidades de grabación de datos en cartucho son exactamente las mismas que las de impresión en pantalla o impresora. Con esto queremos decir que es posible incluir literales, variables de cadena, numéricas y matrices de ambos tipos. El formato general de la sentencia es:

PRINT #n,lista-expresiones

Donde n representa el número de canal, y por lista de expresiones entendemos cualquier clase de

En el sistema secuencial es necesario efectuar la lectura de todos los registros anteriores para tener acceso a uno determinado.



¡!

Los ficheros de datos han de abrirse antes de hacer uso de ellos tanto para lectura como escritura, así como cerrarse una vez concluida la consulta o actualización de los mismos.

*

El sistema utiliza los canales 0 y 1 para la salida de datos hacia la pantalla y para la entrada de caracteres desde el teclado. Los canales 2 y 3 son empleados para producir las salidas de **PRINT** y para **LPRINT** en impresora, respectivamente.

¡!

La unidad MICRODRIVE permite la creación y mantenimiento de ficheros de datos de organización secuencial

*

La sentencia **PRINT #** escribe sobre una zona reservada de la memoria RAM denominada *buffer*

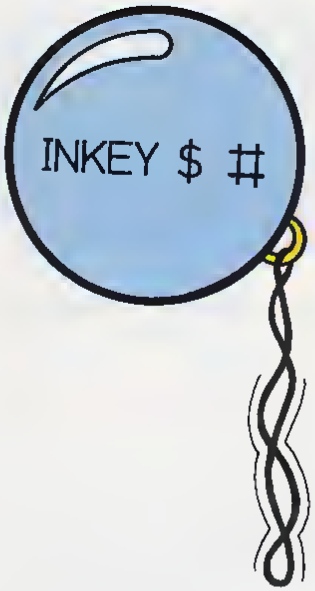
literales o variables de los descritos en el párrafo anterior.

Un elemento a tener muy en cuenta es la obligación de incluir en los lugares oportunos las correspondientes marcas de fin de expresión (un carácter 13 generador de un «retorno de carro»), cuando se incluya más de un elemento a imprimir (literal o variable) por sentencia **PRINT#**.

Para comprender esto con mayor claridad, debemos pensar en que cada sentencia **PRINT** coloca la lista de expresiones que le siguen, en la siguiente línea de pantalla a la última en que se realizó la impresión, en el supuesto de no terminar ésta en punto y coma ; Por ello, cuando se incluye en la sentencia **PRINT #** más de un elemento a imprimir, éstos deben separarse con apóstrofes '.

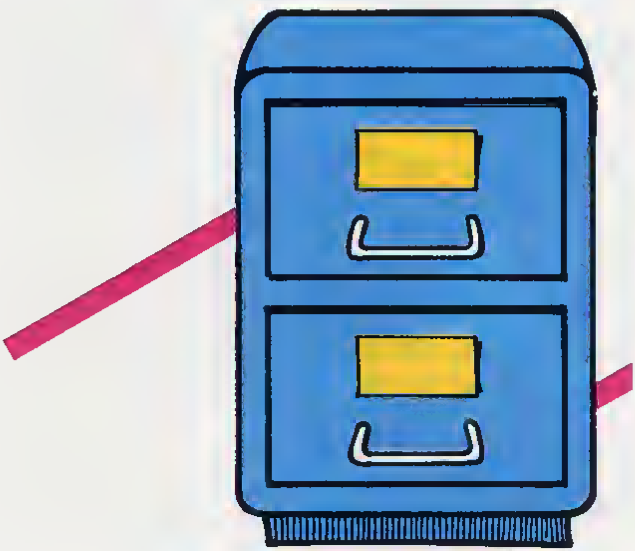
Otra característica que debemos conocer sobre la grabación de datos en cartucho es que no se produce en el momento exacto en el cual es ejecutada la sentencia **PRINT #** Esto se debe a que **PRINT #** escribe sobre una zona reservada de la memoria RAM denominada *buffer*.

El *buffer* es una zona de almacenamiento temporal de datos, donde éstos se van colocando hasta agotar la capacidad total disponible de 512 bytes



Otra forma de lectura de un fichero en cartucho, denominada carácter a carácter, se efectúa con la sentencia **INKEY #**

Llamamos fichero al conjunto de registros relacionados entre sí, guardando un determinado orden y conteniendo información del mismo tipo



Una vez completada ésta, los datos pasan a grabarse en bloque en el fichero abierto en cartucho, para dejar sitio libre a los que vengan detrás de ellos a ocupar su sitio en el *buffer*.

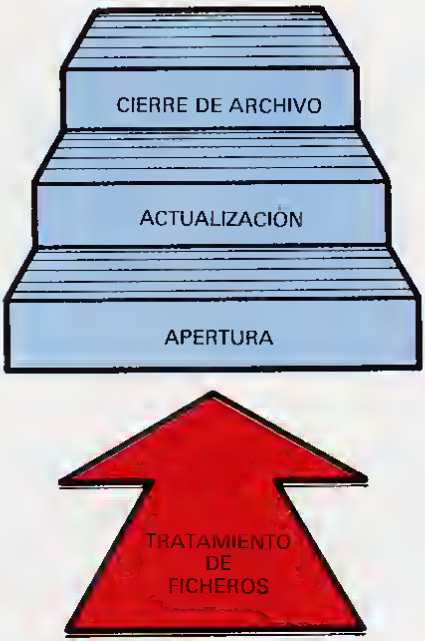
Esta estrategia de proceso de datos se utiliza para reducir el número de accesos al fichero en cartucho, dado que esta operación lleva más tiempo que cualquier otra realizada en la memoria interna del ordenador. Además, da sentido pleno a la sentencia de cierre del fichero

CIERRE DE FICHEROS

La operación de cierre de un fichero se efectúa a través de la sentencia **CLOSE #** El formato general de la sentencia es

CLOSE #c

Donde c indica el número de canal al cual fue asignado el fichero. Al cerrar el canal correspondiente a un fichero abierto, se producen dos acciones: la primera de ellas, la grabación del último bloque de datos contenidos en el *buffer*, esta vez sin esperar a que se complete; y la segunda, la colocación de una marca de fin de fichero con el fin de garantizar el almacenamiento de la información de forma que pueda ser recuperada posteriormente para su lectura.



Cuando tratamos con ficheros en MICRODRIVE es necesario pasar por tres fases: apertura, actualización y cierre de archivo

LECTURA DE DATOS

La lectura de datos en cartucho se efectúa por medio de la sentencia **INPUT #**. Esta sentencia permite la lectura de ficheros en cartucho, siendo necesario especificar a continuación el número de canal por el cual leer los datos. El formato general de la sentencia es:

INPUT #n,lista-expresiones

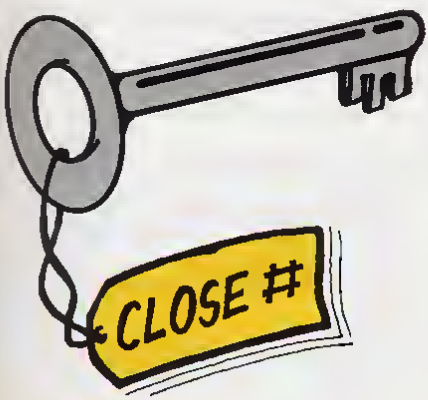
INPUT # acepta el dato o datos incluidos en la lista de expresiones, asignando a cada nueva variable el siguiente valor leído del fichero. Por ello, es importante colocar los caracteres 13 (apóstrofes) en el lugar adecuado en el momento de la grabación. Si por error, grabamos con anterioridad dos variables distintas separadas por un punto y coma ; en vez de los dos puntos :, recuperaremos las dos juntas con una sóla operación **INPUT #**, de forma no deseada.

Existe además otra forma de lectura de un fichero en cartucho, que podemos denominar como «carácter a carácter». Para ello, nos servimos de la sentencia **INKEY$ #**, cuyo formato general es:

INKEY$ #n,lista-expresiones

Con este procedimiento el control de comienzo y final de los datos debe ser gestionado por nosotros de forma manual, al no venir separado cada dato del siguiente por un carácter 13. Sin embargo, aun siendo más lenta que la sentencia **IN-**

El cierre de ficheros se efectúa a través de la sentencia **CLOSE #**



¡!

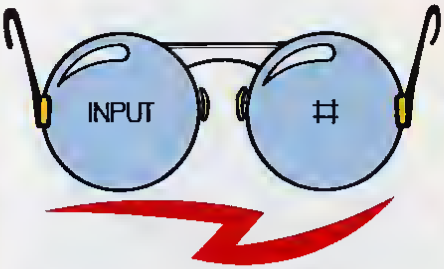
La grabación de datos se hace por medio de **PRINT #**. Esta sentencia permite la escritura en cartucho de forma similar a como lo hace en pantalla su homóloga **PRINT**.

*

La apertura y asignación de nombre y numero de canal a un fichero en MICRODRIVE se produce conjuntamente a través de la sentencia **OPEN #**.

¡!

A pesar de ser el carácter la mínima unidad en que podemos dividir un registro, es el campo la verdadera unidad elemental de información, al ser la porción más pequeña de registro que alberga para nosotros un significado concreto



La lectura en cartucho se efectúa por medio de la sentencia **INPUT #.**

PUT #, puede permitirnos manejos de datos que con la primera no podríamos conseguir

CARACTERES DE CONTROL

Las operaciones de lectura y escritura están regidas por los mismos caracteres de control que operan en la representación de datos en pantalla. Por ello, en lo referente a la sentencia **PRINT #** debemos tener en cuenta las siguientes consideraciones: el punto y coma (;) sirve simplemente como separador sintáctico no imprimiendo nada en el fichero, la coma (,) implica una tabulación a la siguiente mitad de línea, y el apóstrofe (') genera un retorno de carro (nueva línea) separando físicamente los datos.

Por lo que se refiere a **INPUT #**, no está permitido hacer uso de ella refiriéndose a un fichero en cartucho, empleando un literal entrecomillado o variable, como puede hacerse en pantalla, pues ésto produciría el error *Writing to a 'read' file* (escritura sobre un fichero de lectura), debe tenerse también en cuenta que, cuando se almacenan cadenas de caracteres conteniendo comillas ("), la sentencia **INPUT #** las interpreta como fin de la cadena, por lo que es preferible en estos casos el empleo del parámetro **LINE**.

Un problema que puede surgirnos al leer un fichero almacenado en cartucho es el de la detección del final del mismo. Lo cierto es que el ordenador controla esta situación emitiendo el mensaje de error *End of file* al intentar leer el próximo dato al último contenido en el fichero. Sin embargo, esta medida es un poco tardía, puesto que para detectar el final de fichero nos obliga a detener el programa por error.

Solucionar esta situación es sencillo. Basta con incluir después del último dato grabado con **PRINT #** un carácter determinado, por ejemplo un **STOP**, que nos permitirá la detección del final de fichero al reconocer este control en alguna de las posteriores operaciones **INPUT #**.

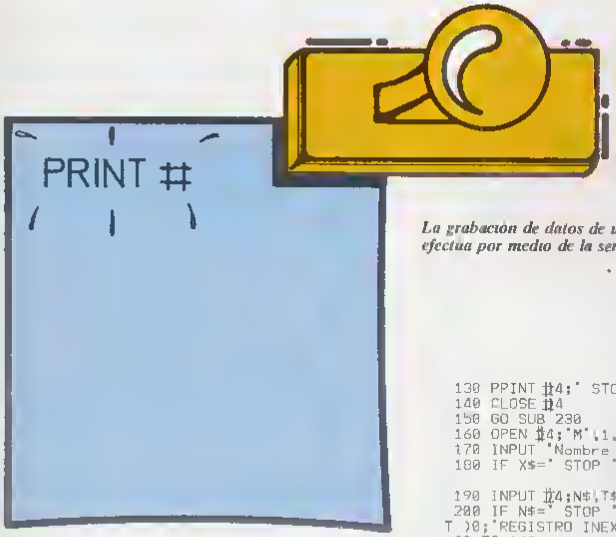
Otra solución consiste en grabar como primer dato del fichero el número de datos que éste contiene: de esta forma, una sencilla estructura **FOR NEXT** nos permitirá recuperar éstos para su lectura, dando valores a la variable contadora desde 1 a la cifra indicada como primer dato del fichero.

PRINT # *escribe sobre una zona reservada de la memoria RAM denominada* **buffer.**

OTRAS APLICACIONES

Existe la posibilidad de almacenar listados de programas en el formato de fichero de datos en cartucho. Este sistema puede servirnos para realizar algún tipo especial de listado, con el empleo de **INKEY$ #**, o para efectuar búsquedas dentro del área de texto BASIC, etc. Para ello, basta con escribir

```
10 OPEN #n;"m";"nombre-fichero"
20 LIST #n
30 CLOSE #n
```



La grabación de datos de un fichero en cartucho, se efectúa por medio de la sentencia **PRINT #**

Otra posibilidad que nos brinda el cartucho es la de efectuar un catálogo del mismo (**CAT n**) grabando su contenido en un fichero, en vez de realizarlo por pantalla. Con ello, se nos permite utilizar esta información por programa.

UNA AGENDA DE TELEFONOS

Con el programa que reproducimos a continuación nos podemos hacer una idea más concreta de cómo manejar los ficheros en cartucho tanto para las operaciones de lectura como de escritura.

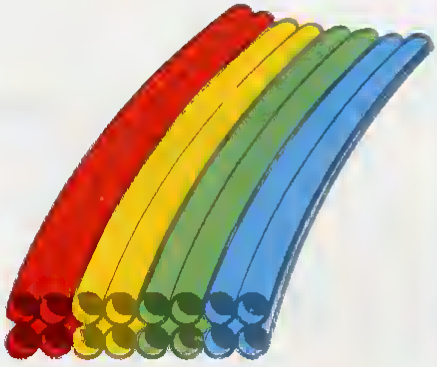
```
  10 REM   AGENDA TELEFONICA (C)J M Lope
z Martinez
  20 LET S$="" : FOR I=0 TO 19 : LET S$=S$
+" " : NEXT I
  30 GO SUB 230
  40 OPEN #4,"M",1,"TELEFONOS"
  50 INPUT "Nombre ", LINE N$: PRINT (N$
+S$) TO 20);" ";
  60 IF N$="STOP" THEN GO TO 130
  70 INPUT "Telefono ", LINE T$: PRINT (
T$+S$)( TO 12)
  80 PRINT #0,"Correcto (S/N)?"
  90 PAUSE 0: LET X$=INKEY$: IF X$<>"S"
AND X$<>"N" THEN GO TO 90
 100 INPUT "": IF X$="N" THEN GO TO 50
 110 PRINT #4;N$;T$
 120 GO TO 50
```

```
 130 PRINT #4;"STOP": "STOP"
 140 CLOSE #4
 150 GO SUB 230
 160 OPEN #4;"M",1,"TELEFONOS"
 170 INPUT "Nombre ", LINE X$
 180 IF X$="STOP" THEN CLOSE #4: STOP

 190 INPUT #4;N$,T$
 200 IF N$="STOP" THEN CLOSE #4: PRIN
T 20;"REGISTRO INEXISTENTE": BEEP 1,-20
 GO TO 160
 210 IF X$=N$ THEN PRINT (N$+S$)( TO 20
),(T$+S$)( TO 12) CLOSE #4: GO TO 160
 220 GO TO 190
 230 CLS : PRINT "N O M B R E";TAB (20),
"TELEFONO"
 240 FOR I=0 TO 31: PRINT "=";: NEXT I
 250 RETURN
```

El programa se divide en cuatro bloques fundamentales: el primero es el de inicialización de variables (línea 20), el segundo es el de entrada de nombres y teléfonos (líneas 30-140), el tercero el de consulta de los datos grabados en el fichero (líneas 150-220) a través del anterior bloque de programa y el cuarto es el compuesto por el área de subrutinas (líneas 230-250).

El Spectrum dispone de 16 canales por los cuales puede circular la información de forma independiente



¡!

El Spectrum dispone de 16 canales (0-15) por los cuales puede circular la información de forma independiente.

*

La información contenida en los registros se encuentra dividida en campos a su vez compuestos por una serie de caracteres (bytes).

*

Tanto el casete como la unidad MICRODRIVE nos permiten almacenar programas BASIC, bloques de memoria, pantallas y matrices de datos.

Para detener la lectura de un fichero en cartucho una vez llegado el final del mismo, debemos incluir después del último dato grabado el carácter **STOP**, *o cualquier otro que fijemos de antemano*

¡!

INPUT # acepta el dato o datos asignando a cada nueva variable el siguiente valor leído del fichero.

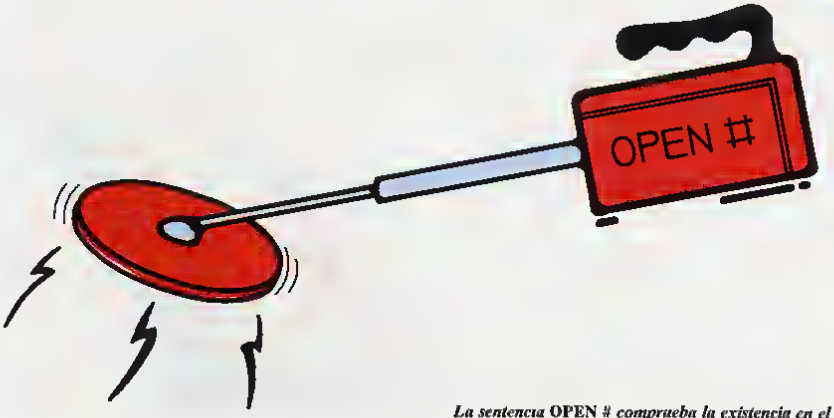
*

También puede leerse un fichero carácter a carácter por medio de **INKEY$ #**

*

Un fichero es un conjunto de registros relacionados entre sí, guardando un determinado orden y conteniendo información del mismo tipo

En la línea 20 se define la variable **S$** con un contenido de 20 espacios para formatear las salidas a pantalla
La línea 30 invoca a la subrutina de cabeceras
La línea 40 abre para su escritura el fichero **TELEFONOS** en el MICRODRIVE 1, asignando para su uso el canal 4. La línea 50 acepta e imprime los nombres. La línea 60 efectúa un control para permitir el fin de la entrada de datos, el cual se producirá cuando en el campo del nombre se teclee **STOP (SYMBOL SHIFT** y **A)**
La línea 70 acepta e imprime los números de teléfono. La línea 80 imprime en la última línea de pantalla un mensaje de validación de los datos tecleados, presentando las opciones **S** y **N**. La línea 90 acepta, haciendo uso de **PAUSE 0** e **INKEY$**, un carácter del teclado.
La línea 100 bifurca el programa a la petición de los datos, si la validación resulta ser negativa. La línea 110 escribe los datos (nombre y número de teléfono) en el fichero número 4, haciendo uso del apóstrofe como separador de campos. La línea 120 cierra el ciclo de petición de datos con un salto a 50. La línea 130 hace una marca de fin de fichero escribiendo los dos datos como caracteres **STOP**. Por último, la línea 140 cierra el canal número 4.
La línea 150 llama a la subrutina de cabeceras.
La línea 160 abre nuevamente el fichero, esta vez para lectura. La línea 170 acepta el nombre a buscar en el fichero. La línea 180 controla la salida de programa por la entrada en el campo del nombre de un carácter **STOP**.
La línea 190 lee las dos variables **N$** y **T$** del fichero en cartucho. La línea 200 controla el final de fichero, imprimiendo el mensaje de registro inexistente. La línea 210 controla si se ha encontrado el nombre buscado bifurcando, en caso afirmativo a la petición de datos, después de imprimir el número de teléfono correspondiente. Por último, la línea 220 cierra el ciclo de búsqueda de datos.
Las líneas 230-240 imprimen la cabecera con los nombres de los campos y un subrayado de guiones.



La sentencia **OPEN #** *comprueba la existencia en el cartucho de un archivo con el nombre que indiquemos*

DISCOS FLEXIBLES

Uno de los momentos cumbre experimentados por todo usuario, cuyo primer contacto con la informática haya sido su Spectrum, es aquel en que por fin, la caja que lo contiene, se encuentra ante él. Apresurádamente, procede a su desembalado, su conexionado, y cuando finalmente el ordenador da señales de vida, mostrando en la pantalla del televisor el mensaje de *copyrigth* del fabricante, todo el poder de los pequeños ordenadores personales parece rendirse ante el usuario.

Quizá, algunos recuerden en esos momentos, los objetivos que primaron a la hora de decidir la compra del Spectrum: podría servirme como agenda, limpiaré el espacio de marcianos, ordenaré de una vez los libros de la biblioteca, mantendré una estricta contabilidad doméstica o clasificaré mi abultada discoteca.

En este punto, solo quedan dos posibilidades: la primera consiste en adquirir en el mercado programas capaces de solventar estas actividades, decisión poco creativa, pero práctica; la segunda es construirnos esos programas a nuestra medida.

Si nuestra elección fue la última, tanto si tenemos como si no conocimientos de programación, comenzaremos a practicar o realizar nuestro programa tecleándolo y consecuentemente, almacenándolo en la memoria. Es entonces cuando surge la necesidad de un nuevo equipo de los conocidos como periféricos u opcionales, capaz de salvar la información contenida en nuestro micro, de manera que no sea necesaria volver a teclearla cada vez que lo desconectemos. Recordemos que ésta se pierde al desenchufar, pues las memorias empleadas en el Spectrum para el almacenamiento de datos y programas, así como en la mayoría de los ordenadores domésticos, exceptuando algunos equipos portátiles, son de las llamadas «volátiles».

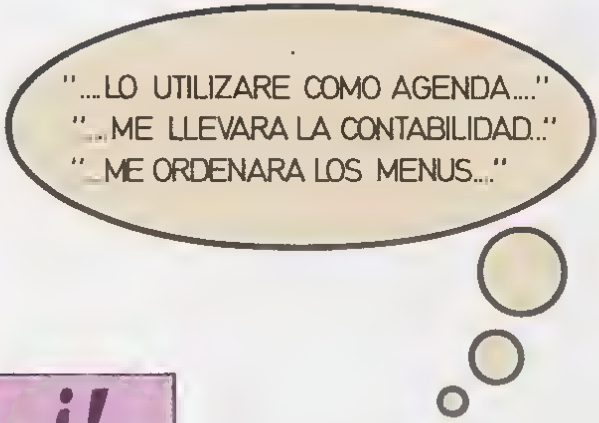
En principio, un sencillo casete portátil, parece la solución adecuada. Pero con cierta desilusión y una buena dosis de susceptibilidad, comenzamos a sospechar que la informática no es tan rápida como se dice. Un programa de tamaño medio, es capaz de exasperar al más paciente de los mortales, cada vez que aguardamos varios minutos hasta completar su carga o grabación.

Los usuarios de ordenadores Sinclair, tienen la posibilidad de completar su equipo con el ZX Microdrive. La velocidad en los procesos de lectura/escritura de información, aumentan considerablemente respecto al sistema de casete. Pero el operador habitual de este periférico puede señalar junto a esta indudable ventaja, dos inconvenientes principales: su baja capacidad de almacenamiento (un cartucho puede llegar a contener entre 85 y 95 Kbytes de información, por término medio), y su escasa fiabilidad.

El escalón siguiente lo constituyen las unidades de disco flexible. Pero sobre estos dispositivos, existe un maremagnun de capacidades, tamaños, tanto de la unidad como del disco, velocidades de transmisión de los datos, sistemas de manejo, etc. que enturbian la posible elección de uno de estos equipos por parte del futuro usuario, el cual se guía en muchas ocasiones más por la publicidad, que por un análisis exhaustivo de sus posibilidades. Emplear discos con el objetivo de conservar programas capaces de realizar un holocausto de naves espaciales, dista mucho de las potenciales aplicaciones de estos periféricos.

¿UNIDAD DE DISCO?

Normalmente, cuando hablamos de unidad de disco, no nos referimos a un único dispositivo el



Seguramente a la hora de decidir la compra de nuestro ordenador entre los objetivos que primaron fueron: podría servirme como agenda, ordenare los libros de la biblioteca

¡!

Una precaución a tomar con los discos flexibles consiste en evitar que se doblen, pues esto ocasionaría un perjuicio irreparable.

*

Como en cualquier soporte magnético las zonas de este material que no se encuentran protegidas no deben ser manipuladas en modo alguno

cual se conecta directamente a nuestro Spectrum, sino a una serie de ellos, los cuales conjuntamente posibilitan el funcionamiento del equipo Aunque según los modelos, varios elementos pueden ir integrados bajo la misma carcasa, son cinco los que configuran el sistema

— el propio disco flexible,
— el dispositivo de lectura/escritura,
— el controlador,
— la fuente de alimentación,
— un interface para la conexión con el ordenador

Comencemos por el final el interface es el elemento dedicado a recoger todo el protocolo de datos de entrada/salida entre el ordenador y la unidad de disco Por ello este elemento debe estar específicamente diseñado para funcionar con el Spectrum, pues en otro caso, aparte del peligro de averías, por más que nos empeñemos, la unidad será inservible

La fuente de alimentación es la encargada de suministrar las tensiones necesarias, para el correcto funcionamiento de todos los módulos Nuevamente, esta es exclusiva de cada unidad de disco, y si no respetamos los consejos del fabricante, nos arriesgamos a pagar caro el uso de una diferente a la establecida

El controlador es el elemento que media entre el ordenador y la unidad de disco, de forma tal que los datos recogidos o enviados a través del interface puedan ser escritos o leídos en el disco, según las órdenes que este elemento establece

El dispositivo de lectura/escritura (*drive*, en su denominación inglesa), tiene como misión grabar o recoger del disco los datos especificados Es el elemento en el cual, mediante una ranura efectuada normalmente en la parte frontal, introducimos el disco para su utilización Un motor colocado en su interior suministra las revoluciones necesarias para hacer girar el disco

Además, en su interior hallamos otros componentes como el cabezal de lectura/escritura, el cual puede tener una o dos cabezas para trabajar con discos de una o dos caras Otros elementos como el detector de protección contra escritura o el detector del hueco índice (más adelante veremos que es), configuran la arquitectura principal de este dispositivo

El disco flexible, disquete o *floppy* (flojo o flexible), se le conoce bajo las tres denominaciones anteriores, es una delgada corona circular de material plástico, normalmente vinilo o mylar, recubierta de compuestos magnéticos a base de oxidos de diversos elementos (oxido de hierro es lo común), tales como los utilizados en las casetes convencionales, cuyo fin es el de servir de medio de acopio masivo de información

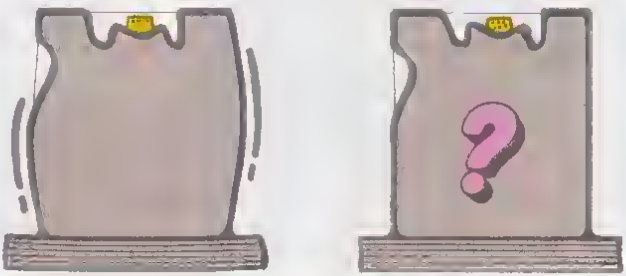
Su aspecto real queda enmascarado, al estar encerrado dentro de una cubierta protectora de forma cuadrada, la cual va forrada en su superficie interior por una capa de material antifricción y antiestático que facilita el giro del disco dentro de su envoltura

Sobre la cubierta protectora, generalmente se han practicado cuatro diferentes perforaciones, las cuales responden a distintos objetivos El orificio más grande, el central, está destinado a permitir el ajuste del dispositivo de arrastre encargado de hacer girar el disco dentro de su funda

El alargado situado en la parte inferior, tiene como misión permitir el desplazamiento longitudinal de las cabezas de lectura/escritura, de manera que éstas tengan libre acceso a toda la superficie útil del disco

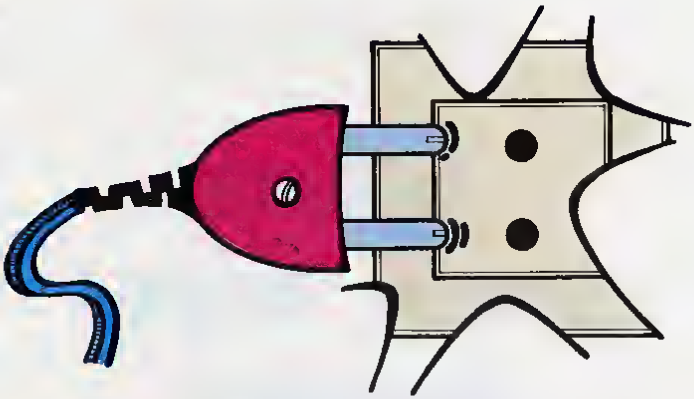
El pequeño orificio situado en las inmediaciones del central es el denominado hueco índice, a partir del cual el sistema es capaz de distinguir el ori

El MICRODRIVE para Spectrum tiene dos inconvenientes principales su baja capacidad de almacenamiento y su escasa fiabilidad



gen de las diferentes secciones en las que el disco está dividido
Finalmente, dos pequeñas hendiduras colocadas en el borde, a ambos lados del hueco de lectura/escritura, aseguran el perfecto ajuste del disquete dentro de la unidad

GUERRA DE FORMATOS



Las memorias empleadas en el Spectrum son llamadas «volátiles», ya que al desenchufar el ordenador perdemos toda la información generada.

Sería ideal, al igual que podemos intercambiar datos y programas entre varios usuarios del Spectrum, tan solo con grabar éstos en un sencillo casete, independientemente de los aparatos que graban o reproducen, hacer lo mismo utilizando como soporte el disco flexible. Desgraciadamente, esto no es posible, de no ser que todos dispusieran de una unidad de la misma marca e incluso más utilizaran discos del mismo tamaño.
Hasta la aparición de los microordenadores eran dos las dimensiones normalizadas en estos soportes: los de 8" y los de 5.25". Pero la miniaturización en los componentes y la consecuente disminución del tamaño de los ordenadores dio origen a otros soportes más pequeños: 3", 3.25", 3.5" y 4". Incluso la todopoderosa I.B.M. dispone de un formato especial de 3.9"
Si hacemos caso de la publicidad que sobre estos soportes reza, indicando que pueden transportarse dentro del bolsillo de una camisa, parece concluyente su menor ocupación de espacio. En la práctica esto no es así, pues los microdisquetes van protegidos mediante una carcasa rígida de 3 a 5 mm de espesor, mucho más gruesa que la de sus hermanos mayores de 5.25". Pero su empleo, repercutirá en las dimensiones de las unidades de disco, condicionante final del tamaño del ordenador

El ordenador nos da señales de vida cuando nos muestra en la pantalla el mensaje de copyright del fabricante.

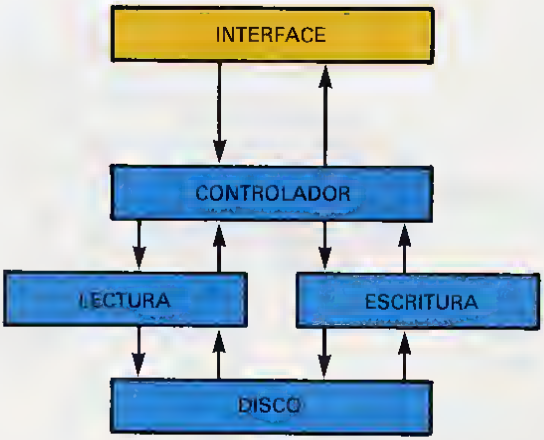
Lo cierto es que los intereses comerciales de las distintas marcas son los que priman a la hora de sembrar el caos entre los posibles usuarios, los cuales, deben reflexionar cuidadosamente la utilización de uno u otro formato, en función de sus necesidades reales. El ANSI (American National Standard Institute, Instituto de Normalizaciones Nacional Americano), ha negado el visto bueno a ciertos modelos de algunas marcas, debido a los distintos espesores tanto de la carcasa protectora como de la capa magnética, pero éstas firman alianzas comerciales con fabricantes de hardware, de forma que se desarrollen *drives* adecuados a este tipo de discos. En resumen, la normalización de dimensiones, no parece cerca.
Otro tema es la cantidad de información que el soporte es capaz de almacenar. Son dos los factores que intervienen: disco de una o dos caras, y simple o doble densidad. En la tabla 2, se dan unos valores orientativos en este sentido, aunque algunos fabricantes investigan en sistemas de registro vertical, que posibilitan el aumento de la capacidad de un disco de 5,25" hasta 10 Mbytes. La densidad de la superficie magnética se mide en TPI (*Tracks per Inch*, pistas por pulgada), es decir, los de doble densidad dispondrán del doble de pistas y, por tanto, duplicarán su capacidad.
Indicar, no obstante, que aunque el disco sea de doble cara y doble densidad, si el lector no está preparado al efecto, por ejemplo, por disponer de una sola cabeza de lectura/escritura, la cual sólo

¡!

Dada la naturaleza plástica del soporte magnético es conveniente mantenerlo alejado de cualquier fuente intensa de calor (estufas, radiación solar, etc.)

*

A efectos de organización es muy conveniente etiquetar los disquetes recien formateados



El controlador media entre el ordenador y la unidad de disco, de forma tal que los datos recogidos o enviados a través del interface se ejecutan según las órdenes que este elemento establece

accede a una de las caras, no conseguiremos obtener rendimiento alguno empleando este formato

EL D.O.S.

El *Disk Operating System*, D.O.S. (sistema operativo de disco) es el segundo de los factores que imposibilita la compatibilidad de disquetes grabados en unidades de diferentes marcas. En lo referente a nuestro Spectrum, cada fabricante proporciona para la gestión del equipo su propio D.O.S.

Este programa es el encargado de aceptar las órdenes de carga y grabación de datos y programas, abrir y cerrar los ficheros existentes, proporcionarnos un catálogo de la información almacenada en el soporte y, en fin, controlar todo el flujo de ordenes y mandatos que afecten a la unidad de disco. Contiene, por lo general, ciertas utilidades, las cuales permiten proteger, buscar ficheros o copiarlos (*back-up*) de un disco a otro, etc.

Antes de trabajar con un disco virgen, es condición indispensable proceder a su formateado. Durante este proceso, el D.O.S. se encarga de dividirlo en pistas y cada una en sectores. El sector es la cantidad mínima de información que puede extraerse o grabarse en el disco. Si el bloque de información no cubre totalmente un sector, es el D.O.S. el encargado de poner automáticamente los bytes restantes a cero. Del mismo modo, si éste es mayor, se ocupan los sectores necesarios para su completo almacenamiento.

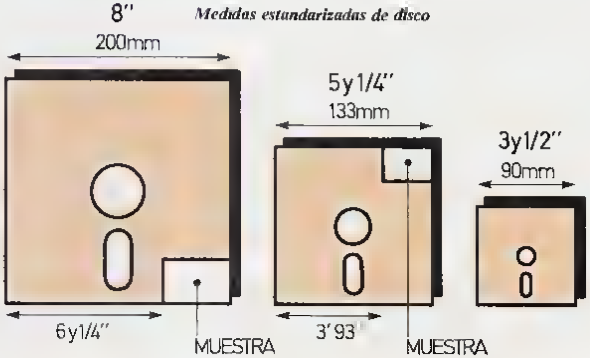
Emplear discos para conservar programas de naves espaciales, dista mucho de las potenciales aplicaciones de estos periféricos

El número de pistas varía según los fabricantes, estando comprendido habitualmente entre 35 y 77 por cada cara útil. Así mismo, la cantidad de sectores es variable, 8, 15, ó 26, por cada pista. Lo mismo ocurre con los bytes por sector: 128, 256 ó 512.

En el momento de dar formato al disco, el D.O.S. numera las pistas de afuera hacia adentro, quedando reservada la primera (00) para el índice general. La unidad de discos identifica cada sector, a través del hueco índice, quedando etiquetado el primero físicamente encontrado a partir de él con 01.

Además, durante la inicialización se crea un campo de identificación para cada sector, el cual contiene información relativa al número de pista en que se encuentra, número del sector, número de cabeza y longitud del sector, además de otros bytes destinados a aumentar la fiabilidad de los procesos de lectura/escritura.

Medidas estandarizadas de disco



La funda envolvente de estos soportes, contiene en su mayoría, recomendaciones indispensables para alargar la vida del disco. En los dibujos se describen algunas de ellas, y es absolutamente indispensable respetarlas, si no queremos condenarlo a la pérdida definitiva de la información contenida.

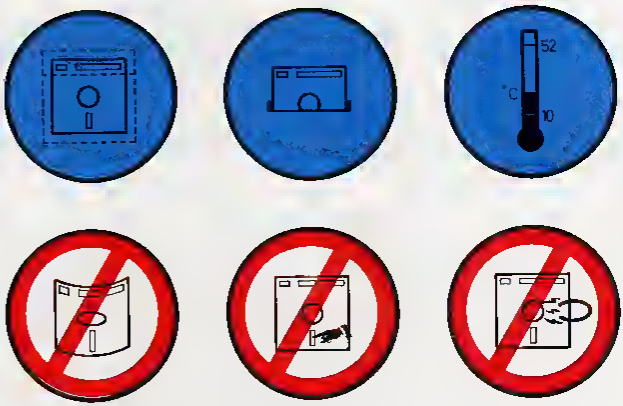
El sistema operativo del disco se encarga también de suministrar las órdenes que activan el motor. Este gira aproximadamente a 300 r.p.m. y durante su movimiento el cabezal se apoya sobre su superficie rastreando las pistas, por lo cual, al cabo de cierto tiempo de uso conviene sustituirlo.

El tiempo de posicionamiento de las cabezas varía de unos equipos a otros, pero no debe ser superior a los 40 ms, mientras que el dedicado a localizar una determinada cabecera de información está por debajo de los 500 ms.

Otro factor a tener en cuenta es la velocidad de transmisión de los datos entre unidad de discos y ordenador, pero por dar una cifra aproximativa y comparativa respecto de la cinta magnética, esta los envía a razón de 200 bytes/s, mientras que un sistema de disco lo hace a 125.000 bytes/s.

El siguiente escalafón tras el disco flexible, son las denominadas unidades de disco rígido. En ellas la velocidad relativa de giro del disco respecto de las cabezas alcanza los 160 km/h. De esta manera se consiguen dos objetivos: mayor velocidad de acceso a la información y mayor duración, puesto que las cabezas prácticamente vuelan sobre la superficie del disco, quedando separadas de él unas micras.

No parece posible, al menos por el momento, ver conectado uno de estos sistemas de disco rígido a un Spectrum, pero la ampliación de nuestro micro con una unidad de disco flexible, potenciará las enormes posibilidades de las que de por sí, disfrutan los usuarios de este ordenador.



Consejos a seguir para mantener nuestros discos en buen estado.

CARA	DENSIDAD	CAPACIDAD	FORMATO
S	S	0,25 MB	8"
S	D	0,5 MB	8"
D	S	0,5 MB	8"
D	D	1 MB	8"
S	S	160	5,25
S	D	320	5,25
D	S	320	5.25
D	D	640	5,25
S	S	160	3 ½"
S	D	320	3 ½"
D	S	300	3 ½"
D	D	640	3 ½"

S = Simple
D = Doble

	CAPACIDAD	ACCESO	AMPLIACION	FORMATO SOPORTE
MICRODRIVE	85/95 FORMATEADO	S	HASTA 8	CARTUCHO CINTA SIN FIN
WAFADRIVE	16/64/128 FORMATEADO	S	NO	CARTUCHO CINTA SIN FIN
INVESDISK	160 K/640 SIN FORMATEAR	S/A	HASTA 4	3 ¼"
TECHNOLOGY RESEARCH	100/160/400/640	S/A	HASTA 4	5,25"
DISCOVERY	180/720 K	S/A	NO	3 ½"

S = Secuencial
A = Aleatorio

¡!

Nunca escribamos la etiqueta del disquete una vez adherida, dado que ello puede deteriorar la superficie magnética albergada en el interior.

*

Para evitar la escritura accidental en un disquete debemos obturar la abertura que presenta la funda protectora, mediante un adhesivo («mariposa») que el fabricante suministra junto con las etiquetas.

LAS CUATRO EN RAYA

Puesto que cualquiera de nosotros hemos puesto en práctica nuestra habilidad mental con este dinámico juego, no creemos necesaria una detallada relación de las reglas. En cualquier caso, para los olvidadizos y los no iniciados, aquí está la descripción somera del juego.

REGLAS DE JUEGO

Dos jugadores, dos grupos de fichas cada uno de distinto color, y un tablero de juego son los elementos indispensables para desarrollar correctamente una partida.

Para vencer, cualquiera de los dos jugadores debe alinear cuatro fichas de su color en línea recta, bien sea en sentido horizontal, vertical o diagonal. En turnos alternativos, cada uno de los contendientes introducirá una ficha propia. Como el árbitro imparcial e insobornable va a ser nuestro inseparable amigo Spectrum, deberemos indicarle el número de columna en la que queremos introducir la ficha en cuestión.

EL PROGRAMA

El juego comienza con una primera pantalla de captación de datos, a través de la cual se piden los nombres de los oponentes cuyo ingenio y habilidad van a entrar en liza. Dichos nombres, tal y como refleja el mensaje visualizado, no pueden exceder la longitud de nueve caracteres.

Una vez realizada esta operación, el ordenador diseña el tablero de juego, asigna un color de ficha a cada jugador, y determina aleatoriamente a qué oponente le corresponde realizar la primera jugada.

Cada vez que uno de los jugadores introduce una ficha, el programa comprueba si ha completado un conjunto de cuatro fichas alineadas. Cuando alguno de los contendientes lo consigue, es decir, ha dispuesto cuatro en raya, aparecerá en la pantalla el nombre del flamante vencedor. Posteriormente el Spectrum nos preguntará si deseamos empezar una nueva partida. Basta pulsar la

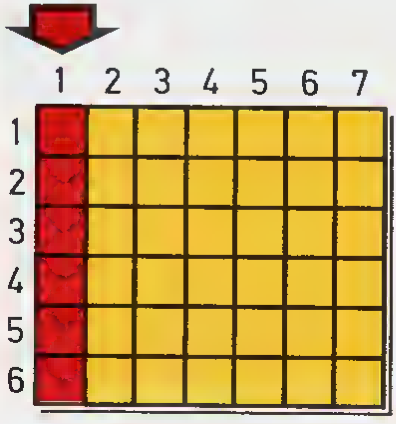
¡!

Es conveniente almacenar el programa antes de su ejecución, para ello hay que recurrir al comando **SAVE** con el siguiente formato **SAVE "4 EN RAYA"**. Si la grabación del programa la quisieramos con auto-ejecución optaríamos por el siguiente formato del comando, **SAVE "4 EN RAYA" LINE 10**

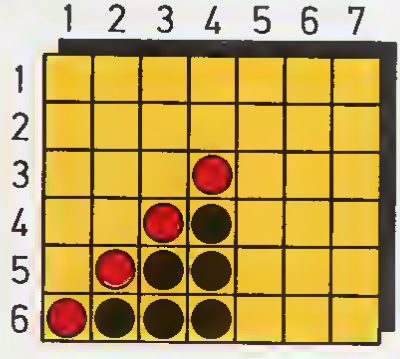
*

Para efectuar un movimiento, debemos indicar al ordenador el número de la columna por la que queremos introducir nuestra ficha

Al programa debemos indicarle la columna en la cual deseamos introducir la ficha en cuestión



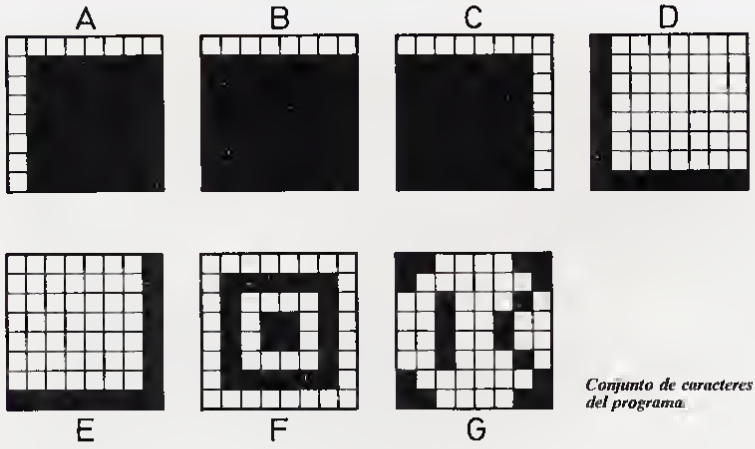
Para vencer, cualquiera de los dos jugadores debe alinear cuatro fichas de su color en línea recta, bien sea en sentido horizontal, vertical o diagonal.



tecla "S" en caso afirmativo o la "N" si la revancha se pospone para mejor ocasión
El programa posee un control sobre las columnas, evaluando cualquier posible error en la introducción de fichas

INTRODUCCION DEL PROGRAMA

Como es habitual, los gráficos definidos se representan en el listado, como el subrayado del carácter cuya tecla se ha de pulsar para la obtención del mismo, en modo **GRAPHICS (CAPS SHIFT + 9)**.
Encontraremos este tipo de gráficos en las líneas 90, 120, 130 y 650. Es conveniente hacer notar, por otra parte, que en la línea 130, entre los gráficos de la **D** y la **E** figura un carácter de subrayado estándar **(SYMBOL SHIFT + 0)**, que no debemos de confundir con la notación gráfica.
Por último, podemos aclarar también que el símbolo que aparece en la línea 720 en la palabra **PRINT** y el número **0**, es un *hush* ( ), que nos permite la escritura en el canal de la parte inferior de la pantalla.
La grabación del programa se conseguirá por el sistema habitual **SAVE "4 EN RAYA"**, o bien **SAVE "4 EN RAYA" LINE 10**.



Conjunto de caracteres gráficos del programa

```
 10 REM ************************
 20 REM * J.M.MAYORAL SERRAND *
 30 REM ************************
 40 REM * 4 EN RAYA  (C) 1985 *
 50 REM ************************
 60 POKE 23658,8
 70 LET CONT=0
 80 LET X=1
 90 LET I$="GGG"
100 INK 0: PAPER 7: CLS
110 DIM A(12,13)
120 LET S$="A8C"
130 LET T$="O_E"
140 LET U$="
150 DATA 0,127,127,127,127,127,127,127
160 DATA 0,255,255,255,255,255,255,255
170 DATA 0,254,254,254,254,254,254,254
180 DATA 128,128,128,128,128,128,128,255
190 DATA 1,1,1,1,1,1,1,255
200 DATA 0,126,66,90,90,66,126,0
210 DATA 195,129,34,36,36,34,129,195
220 FOR J=144 TO 150
230 FOR K=0 TO 7
240 READ A
250 POKE USR CHR$ J+K,A
260 NEXT K
270 NEXT J
280 FOR J=2 TO 17 STEP 3
290 FOR K=2 TO 26 STEP 4
300 PRINT  INK 6; PAPER 0;AT J,K;S$
310 PRINT  PAPER 6;AT J+1,K;T$
320 NEXT K
330 NEXT J
340 PRINT  INK 3;AT 0,3;"1   2   3   4   5   6   7"
350 IF CONT=42 THEN  GO TO B50
360 PRINT  INK 7; PAPER X;AT 20,10;" JUGADOR ";X;" "
370 PRINT  PAPER 6; INK 3;AT 21,2; FLASH 1;" INTRODU
CE NUMERO DE COLUMNA "
380 LET Q$=INKEY$
390 IF Q$="" THEN  BEEP .01,40: GO TO 380
400 IF CODE Q$<49 OR CODE Q$>55 THEN  BEEP 1,-10: GO
 TO 380
410 BEEP .3,40
420 LET A=VAL Q$
430 IF A<>INT A OR A>7 OR A<=0 THEN  GO TO 370
440 LET C=(A-1)*4+2
450 IF ATTR (2,C)<>6 THEN  PRINT  FLASH 1;AT 21,0;"
   COLUMNA ";A;" ESTA COMPLETA    ": BEEP 2,-10: GO S
UB 830: GO TO 370
460 PRINT AT 21,0;"
470 LET CONT=CONT+1
480 FOR J=0 TO 18
490 LET Z=ATTR (J,C)
500 PRINT AT J,C; PAPER X; INK 7;I$
510 IF J=2 THEN  PRINT  INK 3;AT J-2,C;CHR$ 32;A;CHR
$ 32
520 IF J>2 AND Z=6 THEN  PRINT  PAPER 6;AT J-2,C;T$
530 IF J>2 AND Z=48 THEN  PRINT AT J-2,C;"
540 IF J>2 AND Z=56 THEN  PRINT  INK 6; PAPER 0;AT J
-2,C;S$
550 IF J=18 OR ATTR (J+2,C)=23 OR ATTR (J+2,C)=15 TH
EN  BEEP 0.1,-20: GO TO 570
560 NEXT J
570 LET L=J/3+3: LET C=A+3
580 LET A(L,C)=X
590 FOR J=-3 TO 0
600 IF A(L+J,C)=X AND A(L+J+1,C)=X AND A(L+J+2,C)=X
AND A(L+J+3,C)=X THEN  GO TO 690
610 IF A(L,C+J)=X AND A(L,C+J+1)=X AND A(L,C+J+2)=X
AND A(L,C+J+3)=X THEN  GO TO 690
620 IF A(L+J,C+J)=X AND A(L+J+1,C+J+1)=X AND A(L+J+2
,C+J+2)=X AND A(L+J+3,C+J+3)=X THEN  GO TO 690
630 IF A(L-J,C+J)=X AND A((L-J)-1,C+J+1)=X AND A((L-
J)-2,C+J+2)=X AND A((L-J)-3,C+J+3)=X THEN  GO TO 690
640 NEXT J
650 LET I$="FFF": IF X=2 THEN  LET I$="GGG"
660 LET X=X+1
670 IF X=3 THEN  LET X=1
680 GO TO 350
690 PRINT  OVER 1; FLASH 1;AT 20,11;U$( TO 8)
700 PRINT AT 20,9; FLASH 1; INK X;"JUGADOR ";(X);" V
ENCE": FOR A=1 TO 5: FOR B=50 TO 0 STEP -1: OUT 254,B
: NEXT B: BEEP .01,45: NEXT A
710 BORDER 7
720 PRINT #0; FLASH 1;"  OTRA PARTIDITA ?  "; FLASH
0;"  (S/N)"
730 LET A$=INKEY$
740 IF A$="" THEN  GO TO 730
750 IF A$="S" THEN  RUN
760 IF A$="N" THEN  GO TO 10000
770 GO TO 730
780 PRINT AT 20,2;"
790 GO TO 280
800 PRINT AT 21,0;U$;AT 21,16;U$
810 DIM a(12,13)
820 GO TO 280
830 PRINT AT 21,0;"
840 RETURN
850 REM   TABLAS
860 FOR F=10 TO 21
870 PRINT AT F,0;"
880 NEXT F
890 LET A$="COMO PODEIS SER TAN BRUTOTES"
900 PRINT  FLASH 1;AT 12,10; INK 5;"T A B L A S"
910 FOR F=1 TO LEN A$
920 PRINT AT 15,F+1;A$(F)
930 BEEP .05,F+20
940 NEXT F
950 GO TO 720
```